Curs 10: Program ilustrativ iniţial

```cpp
#include<iostream>
using namespace std;
const int dim = 100;

void afiseazaLitDublata(char text[dim], char lit) {
	char ch;
	for (int i = 0; i < dim && (ch = text[i]) != '\0'; i++) {
		cout << ch;
		if (ch == lit) cout << ch;
	}
	cout << endl;
}


void dubleazaLitCuAux(char text[dim], char lit) {
	//dubleaza orice aparitie a caracterului lit
	char ch, aux[dim] = {};
	//formam in aux stringul nou
	for (int i = 0, j = 0; i < dim && (ch = aux[j] = text[i]) != '\0'; i++, j++) {
		if (ch == lit) aux[++j] = lit;
	}
	//mutam aux in text
	for (int i = 0; i < dim && (text[i] = aux[i]) != '\0'; i++) {}
}


void dubleazaLitCuMersInapoi(char text[dim], char lit) {
	int lg = 0;
	for (; lg < dim && text[lg] != '\0'; lg++) {};
	char ch;
	for (int i = 0; i < lg && (ch = text[i]) != '\0'; i++) {
		if (ch == lit) {//prasim din capatul stringului
			lg++;
			for (int j = lg; j > i; j--) {
				text[j] = text[j - 1];
			}
			i++;
		}
	}
}
```

```
void dubleazaLitCuInserare(char text[dim], char lit) {
	char chStg = text[0], chDrp;
	//caut lit cu mana dreapta
	for (int i = 0; i < dim && (chDrp = text[i]) != '\0'; i++) {
		if (chDrp == lit) {
			//am gasit lit
			//chStg = caracterul deja pus in text
			//chDrp = caracterul care urmeaza sa fie pus
			for (int j = ++i; j < dim && chStg != '\0'; j++) {
				chStg = chDrp;		//eliberez mana dreapta
				chDrp = text[j];	//scot din text cu dreapta
				text[j] = chStg;	//si pun cu stanga
			}
		}
	}
}

int main() {
	char text[100] = "abc";
	cout << text << endl;
	//afiseazaLitDublata(text, 'b');
	//dubleazaLitCuAux(text, 'b');
	//dubleazaLitCuMersInapoi(text, 'b');
	dubleazaLitCuInserare(text, 'b');
	cout << text << endl;
}
```

Program ilustrativ final

```
#include<iostream>
using namespace std;

void afisareFaraEgale(char text[dim]) {
	//stringul  "abbbbccaaa" este afisat "abca"
	char ch;
	cout << text[0];
	for (int i = 1; i < dim && (ch = text[i]) != '\0'; i++) {
		if (ch == text[i - 1]) continue;
		cout << ch;
	}
	cout << endl;
}
void afisareFaraEgaleSiFaraStelute(char text[dim]) {
	//stringul  "abb*bbccaaa" este afisat "abca"
	//cautam primul ch diferit de *
	int i = 0;
	while (i < dim && text[i] == '*') i++;
	if (i == dim || text[i] == '\0') return;
```

```
	char chAfisat, ch;
	cout << (chAfisat = text[i]);
	for (i++; i < dim && (ch = text[i]) != '\0'; i++) {
		if (ch == chAfisat || ch == '*') continue;
		cout << (chAfisat = ch);
	}
	cout << endl;
}

void curataEgale(char text[]) {
	//stringul "abbbbccaaa" este transformat in "abca"
	char ch;
	for (int i = 0; i < dim && (ch = text[i]) != '\0'; i++) {
		//stabilim pana unde se repeta ch
		int ii = i + 1;
		while (ii < dim && ch == text[ii]) ii++;
		int delta = ii - i - 1;
		//eliminam delta caractere
		if (delta == 0) continue;
		for (int j = i, jj = i+delta; jj < dim && (text[j] = text[jj]) != '\0';
j++, jj++) {
		}
	}
}

void dubleazaTot(char text[]) {
	//stringul "abc" este transformat in "aabbcc"
	int lg = 0;
	while (lg < dim && text[lg] != '\0')lg++;
	if (2 * lg > dim - 1) return;
	text[2 * lg] = '\0';
	for (int i = lg - 1; i >= 0; i--) {
		text[2 * i + 1] = text[2 * i] = text[i];
	}
}

int main() {
	char text[dim] = "abb*bbccaaa";
	cout << text << endl;
	//afisareFaraEgale(text);
	//afisareFaraEgaleSiFaraStelute(text);
	//curataEgale(text);
	dubleazaTot(text);
	cout << text << endl;
	return 0;
}
```